**3P090**

# スクリプト言語を利用した電子構造計算プロトタイピング環境の開発

(九大情基センター[a], 九州先端研[b], JST-CREST[c])

○本田宏明[a,c], 眞木淳[b], 野呂武司

# A prototyping environment for electronic structure calculation by scripting language

(Kyushu Univ.[a], ISIT[b], CREST-JST[c])

○ Hiroaki Honda[a,c], Jun Maki[b], Takeshi Noro

## 【はじめに】

現在標準的に利用されている種々の分子軌道法計算プログラムでは，SCF から電子相関計算に至るまでの様々な物性計算が利用可能である．しかしながら全体で数百万行に達するプログラムも有り，内部実装の詳細なドキュメントが公開されていない場合が多い．そのためプログラムソース中にのみ記述されている各種データのデータ構造や定義箇所，変更箇所やその条件を正確に把握する事が難しく，密度行列やフォック行列といった基本的なデータについても利用方法を独習し自身のアイデアを実装することは困難であることが多い．このような状況において，実装コード量の減少を期待した Python プログラミング言語による実装 [1,2] も既に公開されており，分子軌道法プログラムの実装の学習を促進させる試みも進んではいるが，現状では電子相関計算において通常の SDCI 計算が可能なプログラムは実装されていない．

これまで我々はスクリプト言語の一つである Ruby を利用し，逐次実行においては対話型プログラムが実行可能であり，並列計算については計算機センターでも利用可能な分子軌道法プログラムを開発してきた [2]．これに対し本研究では，同じく Ruby 言語を利用し CI までを計算可能な逐次実行プログラムを開発することを目的とした．またハミルトニアン行列生成アルゴリズムとして，佐々木による原子に対する表式 [3] を分子の $C_1$ 対称性の系に適用した Configuration State Function (CSF) ベースの多電子行列要素計算表式と，Shavitt による Distinct Row Table (DRT) を組み合わせた方法を用いる．

## 【Ruby 言語を用いたプログラム開発】

Ruby はオブジェクト指向をベースとした動的型付けの汎用プログラミング言語であり，型宣言やメモリの動的確保，開放処理を明示的に記述する必要がなく，組み込みの配列クラスや文字列クラス等の数値計算や文字列の取扱いに必要となる定型の処理が予め用意されている．そのため，他のスクリプト言語と同様に Fortran や C 言語といったコンパイラ言語と比較して記述に必要なプログラムコード量が減少するため，より容易に記述することが可能である．

一方，動的型付けのスクリプト言語による実装では通常のコンパイラ言語によるプログラム実装と比較して種々の原因により実行速度が遅くなる事が知られている．そのためプログラム作成の際に，プログラミング言語由来となる性能低下予想箇所として予期される，1. 配列を利用する箇所については数値計算用配列ライブラリ [4] の利用，2. 行列計算については GSL ライブラリ [4] 等の Ruby からの利用，3. 分子積分や積分変換，ハミルトニアン行列計算に必要な Wigner 係数の計算，Davidson-Liu の対角化プログラムについては，C/Fortran による Ruby からも利用可能なライブラリ作成の工夫を行った．

## 【プログラム構成】

電子相関までの計算を実施することを踏まえ，プログラムは従来型の積分保存型の構成としている．1. 分子積分計算，2.RHF 計算，3. 積分変換，4. エネルギー表式作成，5.CI 計算の計算モジュー

ルの順番で実行し，CI 波動関数とエネルギー，自然軌道が取得可能である．各計算モジュール内では全てオンメモリにて実行しており，それぞれのモジュール間はファイルを利用してデータの受け渡しをする．また，個々のモジュール毎に計算が可能である．

## 【使用アルゴリズムと実装言語】

**分子積分** 分子積分計算に対しては，小原による分子積分に対する一般漸化表式に基づく定式化 [5] を用いた．現状では通常のエネルギー計算に必要となる積分種のみのサポートであるが，今後積分種の増加予定である．C 言語による実装である．

**RHF, UHF** 積分ファイルを利用する通常の RHF, UHF を実装した．Ruby 言語による実装である．

**積分変換** 2 電子積分では 2 つの添字の組合せを組合せた 4 つの添字の三角行列を積分変換する必要がある．今回は AlchemyII プログラム [6] に実装されたアルゴリズムを利用してプログラムを作成した．C 言語による実装である．

**ハミルトニアン行列生成** ハミルトニアン行列のエネルギー表式については，佐々木による tensor-recoupling による定式化 [7] を用いた．これは CSF ベースの方法であり，ハミルトニアン中にスピンに関わる演算子が含まれる場合も計算が可能である．CSF 生成には Shavitt による DRT の方法 [8] を用いた．ハミルトニアン行列要素計算には，上記の方法を利用し，DRT のダイアグラム中を再帰的に探索しつつ，行列要素の存在するループのトップとボトム軌道に共通する全ての演算子分の表式を生成する，Kamuy プログラムに実装されている方法 [9] を用いた．エネルギー表式を求めるための Wigner 係数を除き Ruby 言語による実装である．

**対角化** 対角化には 大次元行列について下から複数個の解を求めることの可能な Davidson-Liu のアルゴリズム [10] を実装した．Ruby 言語と C 言語の両方の実装を利用可能である．

## 【CSF の選択】

ハミルトニアン行列生成の際の CSF スペースの選択のため，SCF 軌道の Frozen, Internal, Active, External, Discarded の指定ならびに，Internal, Active, External については各グループにおける電子占有数を指定可能とした．これにより現状のプログラムにて first-order CI ならびに second-order CI に対応した選択が可能である．
プログラムの開発の進捗ならびに種々のテスト計算，実行時間等については当日報告する．

## 【謝辞】

本研究の一部は JST-CREST の研究課題「省メモリ技術と動的最適化技術によるスケーラブル通信ライブラリの開発」の支援を受けております．

## 【参考文献】

[1] “HOLTON,” [On line]. Available: ⟨https://github.com/theochem/horton⟩.
[2] “PyQuante: Python Quantum Chemistry,” [On line]. Available: ⟨http://pyquante.sourceforge.net/⟩.
[3] 本田 他, 分子科学討論会 2014, 4P110.
[4] “GSL - GNU Science Library,” [On line]. Available: ⟨http://www.gnu.org/software/gsl/⟩.
[5] M.Honda, *J.Chem.Phys.*, Vol.94, pp.3790-3804, 1991.
[6] A.D.McLean *et al.*, *MOTECC-90*, IBM, pp.593-638.
[7] F.Sasaki *et al.*, *MOTECC-90*, IBM, pp.181-234.
[8] I.Shavitt, *Int.J.Quant.Chem.*, Vol.S12, pp.5-32, 1978.
[9] F.Sasaki *et al.*, *Theor. Chim. Acta*, Vol.72, pp.123-138, 1987.
[10] B.Liu, LBNL Paper LBL-8158, UC-32, CONF-780878 (1974).